GRANT/ PR/ IN/ 63

An Investigation of Terminal Guidance and Control Techniques for a Robotic Mars Lander

by
Brian K. Birge
and
Gerald Walberg

Performance Report submitted under
NASA Grant NAG-1-~~2086~~ 2222
for the period 5/15/99 to 5/15/00

North Carolina State University
Mars Mission Research Center

August 23, 2000

Executive Summary:

- developed new baseline case
- new initial M2001 south site handoff conditions
- points used in batch runs encompass total range of possible conditions (75+1 points)
- uses up-to-date Viking engine data
- no control on ballistic chute during baseline
- chute drag converted to body drag for consistency with lifting chute cases at chute jettison
- enhanced data gathering during batch runs to include conditions at parachute jettison
- also saves all trajectories for each batch run in a single 3-dim variable
- This showed that using a constant bank angle to find maximum maneuverability on the chute was a bad idea.
- The constant bank made the trajectories spiral and limited the range we could get.
- developed algorithm to quickly/easily draw ellipses and find centroids and areas of scattered data.
- Changed the non-optimized lifting chute cases such that there is an event that forces the bankangle to zero
- once the azimuth heading is equal to our chosen bankangle+90. This prevented spiralling and maximized the expected range on the lifting chute with respect to bank angle.
- The non-optimized cases were used to find a new surface reference point. I used the visual mean of each different lifting coefficient run (0.5,0.75,1.0,1.5) and averaged those. Lift/Drag ratio of 2.0 and 2.5 had some problems that I'll explain later.
- The new reference surface point for the lifting chute cases is:
  long: 93.8023
  lat: -15.8384
- For the thrusting on a chute cases (one limited to 50kg fuel usage, the other to 62kg) the average is:
  long: 93.7828
  lat: -15.8358
- Non-optimized steering cases just had one control event on the chute, bank for lifting, bank and eta for thrusting.
- The 2.0, and 2.5 L/D trials seemed to have so much lift that when used gammar went to 90. Then the lander would stall and come back down unpredictably oriented making a bankangle choice kind of meaningless from the standpoint of reaching the reference target.
- This is kind of unfortunate as based on the limited parafoil info Eric & I could find it seemed that an L/D of 2.5 would be reasonable.
- Using a limited set of initial conditions and a hands on approach to initial guess choice, results for the hover and lifting parachute cases improved dramatically.

Abstract.

Continuing on previous work, various precision landing control algorithms are examined with the goal of minimizing the landed distance to a specified location on the Mars surface. This study considers a set of points from parachute handoff to touchdown on the surface. The first scenario considers a reverse gravity turn to a hover condition 500 meters above the surface and then uses lateral thrusting to minimize the range to target. The second scenario examines a guided, lifting parachute followed by a powered gravity turn to the targeted landing site. The third scenario considers thrust vectoring while on the ballistic parachute, followed by a reverse gravity turn to touchdown.

Introduction.

For future planetary exploration missions, either robotic or manned, it is desirable to precisely target a lander's touchdown point. Perhaps there has been a previous robotic mission that requires a follow-up robotic mission in order to retrieve collected samples and return them to earth. Perhaps there are specific geographical features that require close-up study. Regardless, a need for precision landing within 100 meters of a specified geographic location exists. Current state of the art can only achieve positioning to within approximately 10 kilometers. This study examines two general methods for controlling a lander as it touches down on a specific point on the Mars landscape. All model parameters and constants are taken from and designed to be compatible with the M2003 specifications when possible.

Nomenclature.

alppc1 first coefficient of alpha equation (POST3D)
alppc2 second coefficient of alpha equation (POST3D)
betpc1 first coefficient of beta equation (POST3D)
betpc2 second coefficient of beta equation (POST3D)
critr criterion used to activate events (POST3D)
depph event at which targeting is to be satisfied (POST3D)
deptl tolerances on dependent variables (POST3D)
depval targets of dependent variables (POST3D)
depvr names of dependent variables (POST3D)
diamp1 diameter of parachute #1 (POST3D)
dprng1 dot product range to reference long, lat (POST3D)
etapc1 throttling parameter polynomial coefficient one (POST3D)
gdalt vertical altitude above oblate planet (POST3D)
gdlat geodetic latitude (POST3D)
idepvr type of constraint desired for dependent variables (POST3D)
ifdeg allows degrees to be used in targeting (POST3D)
iguid(1) guidance desired (POST3D)
iguid(10) separate channel option for pitch (POST3D)
iguid(11) separate channel option for bank angle (POST3D)
iguid(13) relative yaw angle reference option flag (POST3D)
iguid(2) selects either independent or identical channel steering (POST3D)
iguid(9) separate channel option for yaw (POST3D)
indph event that starts perturbing independent variables (POST3D)
indvr names of independent variables (POST3D)
isp specific impulse
ispv vacuum specific impulse (POST3D)
long longitude, degrees or meters (POST3D)
MarsGram program modeling mars atmosphere
Matlab high level programming language
neng number of engines (POST3D)
npc(7) acceleration limit option flag (POST3D)
opt optimization flag (POST3D)
optph optimize by this event (POST3D)
optvar optimization variables (POST3D)
phi angle between due North and target
POST3D Program to Optimize Simulated Trajectories 3D version
rn nose radius
sref aerodynamic surface area
theta angle between due North and velocity vector
ur first component of lander horizontal velocity planet relative (POST3D)
velr vehicle velocity relative to rotating planet (POST3D)
vr second component of lander horizontal velocity planet relative (POST3D)
wgtsg vehicle gross weight (POST3D)
wjett jettisoned weight (POST3D)
wpropi initial propellant weight (POST3D)
wr vertical velocity planet relative (POST3D)

Lander Scenarios.

Scenario 1 – Hover and Thrust Laterally.
At a set of initial (parachute handoff) conditions supplied by LaRC, the lander deploys a ballistic parachute for deceleration. Upon parachute jettison, a reverse gravity turn is performed to achieve a hover condition 500 meters above the Mars surface. At this point lateral thrusting is used to fly the lander to the desired target and achieve desired terminal velocity components. This scenario is discussed in the attachment "Terminal Guidance Techniques for a Mars Precision Lander".

Scenario 2 – Guided, Lifting Parachute.
At a set of initial (parachute handoff) conditions supplied by LaRC, the lander deploys a ballistic parachute to decelerate. Based on the handoff conditions and the desired touchdown point, at some point the ballistic chute is transformed into a lifting steerable parachute. In the real world this is done via 'firewire', some of the parachute vanes are burned off allowing a directed lift. Once the parachute is jettisoned, the lander performs a reverse gravity turn to touchdown.

Scenario 3 – Thrust on Parachute
At a set of initial (parachute handoff) conditions supplied by LaRC, the lander deploys a ballistic parachute for deceleration. During this deceleration phase, lateral thrusting is used to minimize range to target. Upon parachute jettison, a reverse gravity turn is performed to touchdown.

Descent Simulations.

Simulation Algorithms.
The programs used in this study were Matlab, POST3D, and MarsGram. Matlab was used as a shell for controlling the batch runs as well as for visualization and pre/post processing of data. Program To Optimize Simulated Trajectories (POST3D) was used to perform the actual lander simulations on a case by case basis. MarsGram was used to build an atmosphere, which was then converted to tables for use in POST3D.

List of input decks used
m2001newbase2.inp
m2001newbase2-liftchute.inp
m2001newbase2-liftchute-noopt-bnkstudy.inp
m2001newbase2-liftchute-targref.inp
m2001newbase2-liftchute75ld.inp
m2001newbase2-liftchute75ld-noopt-bnkstudy.inp
m2001newbase2-liftchute75ld-targref.inp
m2001newbase2-liftchute1ld.inp
m2001newbase2-liftchute1ld-noopt-bnkstudy.inp
m2001newbase2-liftchute1ld-targref.inp
m2001newbase2-liftchute15ld.inp
m2001newbase2-liftchute15ld-noopt-bnkstudy.inp
m2001newbase2-liftchute15ld-targref.inp
m2001newbase2-liftchute2ld.inp
m2001newbase2-liftchute2ld-noopt-bnkstudy.inp
m2001newbase2-liftchute2ld-targref.inp
m2001newbase2-liftchute25ld.inp
m2001newbase2-liftchute25ld-noopt-bnkstudy.inp
m2001newbase2-liftchute25ld-targref.inp
m2001newbase2-thrustchute.inp
m2001newbase2-thrustchute-noopt-bnkstudy.inp
m2001newbase2-thrustchute12kg.inp
m2001newbase2-thrustchute12kg-noopt-bnkstudy.inp
bc.inp

Most of the input decks follow the same format so there is no need to go through all of them individually. A brief explanation and examination of the basic format using the following decks as templates will suffice:

1) m2001newbase2.inp
2) m2001newbase2-liftchute.inp
3) m2001newbase2-liftchute-targref.inp
4) m2001newbase2-liftchute-noopt-bnkstudy.inp
5) m2001newbase2-thrustchute.inp

6) m2001newbase2-thrustchute-noopt-bnkstudy.inp
7) bc.inp

m2001newbase2.inp
This first input deck is the baseline case. This takes the initial conditions, and pops open the parachute to slow down. There is no thrust applied while the parachute is deployed. Once the parachute is jettisoned, the control system kicks in and performs a reverse gravity turn to touchdown. The lander's touch down latitude and longitude is not targeted but ending velocity and altitude conditions are targeted.

These desired end conditions are:

-1.1<= ur <= 1.1 m/s
-1.1<= vr <= 1.1 m/s
wr = 2.0 m/s
2499 <= gdalt <=2501 m (2500m is surface at landed latitude and longitude)

The above-discussed input deck gives us the baseline landed ellipse. It is desired to shrink the magnitude of this baseline ellipse as much as possible. Before discussing the input decks that attempt to solve that problem, let's look in detail at the baseline case.

Event1 – initial setup and parachute deploy
- Atmosphere input as tables (from a previous MarsGram run)
- MarsGram winds input as tables
- Initial pos/vel input in inertial coordinates from M2001 (now M2003) Monte Carlo analysis
- sref=4.5238934 (aero surface area)
- rn= 0.6638 (nose radius)
- Gravity model uses oblate planet with spherical harmonics j2 through j6
- guidance uses atmospheric relative aerodynamic angles
- wgtsg=2176.811 N (585.479 kg) [vehicle weight including parachute]
- wpropi=372.0 N (100 kg) [initial weight of propellant]
- neng=2, [2 engines]
- ispv(1)=553.9,553.9, [Mars Isp (Earth Isp = 210 sec, mono-propellant hydrazine rocket engine)]
- primary engine is pointed out X body axis
- secondary engine is pitched -90 degrees
- vehicle drag coefficient = 1.7
- parachute drag coefficient = 0.41
- Parachute deployment rate set so full deployment occurs in about 3 sec.
- No optimization is performed

Event22 – parachute diameter limit
- at diampl=13
- wgtsg = 1937.297 N [after dropping heat shield]
- parachute diameter is limited to 13m

Event25 – convert to dragging body
- turn off parachute drag option
- increase lander surface area to 132.73
- change drag of lander to that of parachute
- this event simulates the drag of a ballistic parachute but is event where lifting coefficient will be input for other decks

Event50 – jettison parachute and turn on primary engine (start of reverse gravity turn)
- at gdalt=3500m
- wjett = 276.702 N [weight jettisoned]

- `sref=2.0 [surface area reflects loss of chute]`
- `turn on primary engine and start targeting (using relative aero-angles)`
- `vehicle drag coefficient = 2.0 [increased to reflect non-aero shape of lander]`
- `decrease sref to 2.0 (simulates jettison of lifting parachute)`
- `change back drag of lander to original value`

Event80 – surface touchdown and turn off primary engine
- `critr='wr'`
- `value = 2.0`
- `at this event gdalt is targeted to 2500m`

Event500 – endproblem

m2001newbase2-liftchute.inp

This deck is exactly the same as above except for the following changes:
1) at event 25, a coefficient of lift is input as .402, corresponding to an L/D for the lifting body of 0.5
2) This allows simulation of a lifting parachute.
3) Optimization is performed to minimize the final range to reference landing point
4) 3 more events, 26,27,28 allow the targeting algorithm several events to steer the lander towards reference

The reference point used is:
Latitude: `-15.8384`
Longitude: `93.8023`

This reference point was chosen as a reasonable mean of all reachable points on the lifting parachute domain.

m2001newbase2-liftchute75ld.inp
m2001newbase2-liftchute1ld.inp
m2001newbase2-liftchute15ld.inp
m2001newbase2-liftchute2ld.inp
m2001newbase2-liftchute25ld.inp

...are all the same as described above with the exception of change in L/D for parametric analysis. They correspond to L/D of 0.75, 1.0, 1.5, 2.0, and 2.5 respectively.

m2001newbase2-liftchute-targref.inp

This deck is exactly the same as m2001newbase2-liftchute.inp except instead of targeting to the surface at 2500 meters, it targets 200 meters above (2700m). This allows relaxation of altitude targeting constraint and was considered as having enough distance to perform a specific terminal descent maneuver (TBD). It also has one extra steering event (event 29).

m2001newbase2-liftchute-noopt-bnkstudy.inp

This input deck was used to generate the maximum range for the above targeted cases. It also allowed determination of a reasonable mean target reference in conjunction with the following Matlab programs:

ellipsebanks.m
p3dbatchbankstudy.m

The deck is essentially the same as the baseline case (m2001newbase2.inp) with some minor changes. There is an added event (event 26) that resets bankangle to 0 once certain criteria are met. There is no optimization. Basically what happens is that an initial bank angle is chosen for lifting flight. The lander will start to turn. Once the lander is actually heading has turned the desired amount, the bankangle is reset to 0 and straight gliding flight continues until time to jettison chute and reverse gravity turn to ground. Running this case in batch mode with several initial bankangle commands and several desired headings yields the maximum reachable range for each of the L/D cases. Finding the convex hull of the total maximum set and averaging produced a good target reference.

m2001newbase2-thrustchute.inp

Same as baseline case with the following changes:
1) Optimization is performed to minimize range to reference target
2) Parachute is ballistic
3) Events are structured for 4 engine thrust vectorings (during parachute deployment), events 25,26,27,28
4) 50kg of fuel is saved for the reverse gravity turn powered descent phase

The m2001newbase2-thrustchute12kg.inp deck is the same but allows 12kg more fuel usage for maneuvering. Also, the target reference is different than for the lifting parachute cases:

Latitude: `-15.8358`
Longitude: `93.7828`

The reason for the difference between the reference for thrusting and lifting cases is because the untargeted final position domains are very different.

m2001newbase2-thrustchute-noopt-bnkstudy.inp

Same as the other 'bnkstudy' decks but tailored for the thrusting on the parachute case.

bc.inp

This input deck was used for AIAA paper 2000-53942, Terminal Guidance Techniques for a Mars Precision Lander, and is attached to this report. It is essentially the baseline lifting body case but used MarsGram in real time during the simulation rather than MarsGram generated tables.

Controls.
For all lifting parachute cases except bc.inp, there is a common set of controls used for the reverse gravity turn.

```
c - general Independent Control inputs
      indph(1) = 25,
      indph(2) = 26,26,
      indph(4) = 27,27,
      indph(6) = 50,50,
      indph(8) = 28,28,
c
      indvr(1) = 'bnkpcl',
      indvr(2) = 'bnkpcl','critr',
      indvr(4) = 'bnkpcl','critr',
      indvr(6) = 'etapcl','alppcl',
      indvr(8) = 'bnkpcl','critr',
c
*include '../../bankangleguess.dat',  / just covers u(1)-u(5), i.e. chute events
      u(6)     = 1.0,0.0,
      u(8)     = 0.0,10.0,
```

For the thrusting on the parachute cases:

```
c - general Independent Control inputs
      indph(1) = 25,
      indph(2) = 26,26,
      indph(4) = 27,27,
      indph(6) = 50,50,
      indph(8) = 25,26,27,
      indph(11)= 28,28,28,
c
      indvr(1) = 'bnkpcl',
      indvr(2) = 'bnkpcl','critr',
      indvr(4) = 'bnkpcl','critr',
      indvr(6) = 'etapcl','alppcl',
```

```
     indvr(8) = 'etapcl','etapcl','etapcl',
     indvr(11)= 'bnkpcl','critr','etapcl',
c
*include '../../bankangleguess.dat',  / just covers u(1)-u(5), i.e. chute events
     u(6)      = 1.0,0.0,
     u(8)      = 1.0,1.0,1.0,
     u(11)     = 0.0,10.0,1.0,
```

For bc.inp (discussed in attachment) the controls are:

```
c - general Independent Control inputs
     indph     = 50,50,50,50,
                 70,70,70,
     indvr     = 'etapcl','etapc2','alppcl','betpcl',
                 'etapc2','alppcl','betpcl',
     u         =1.0,0.0,90.0,0.0,
                 0.0,90.0,0.0,
```

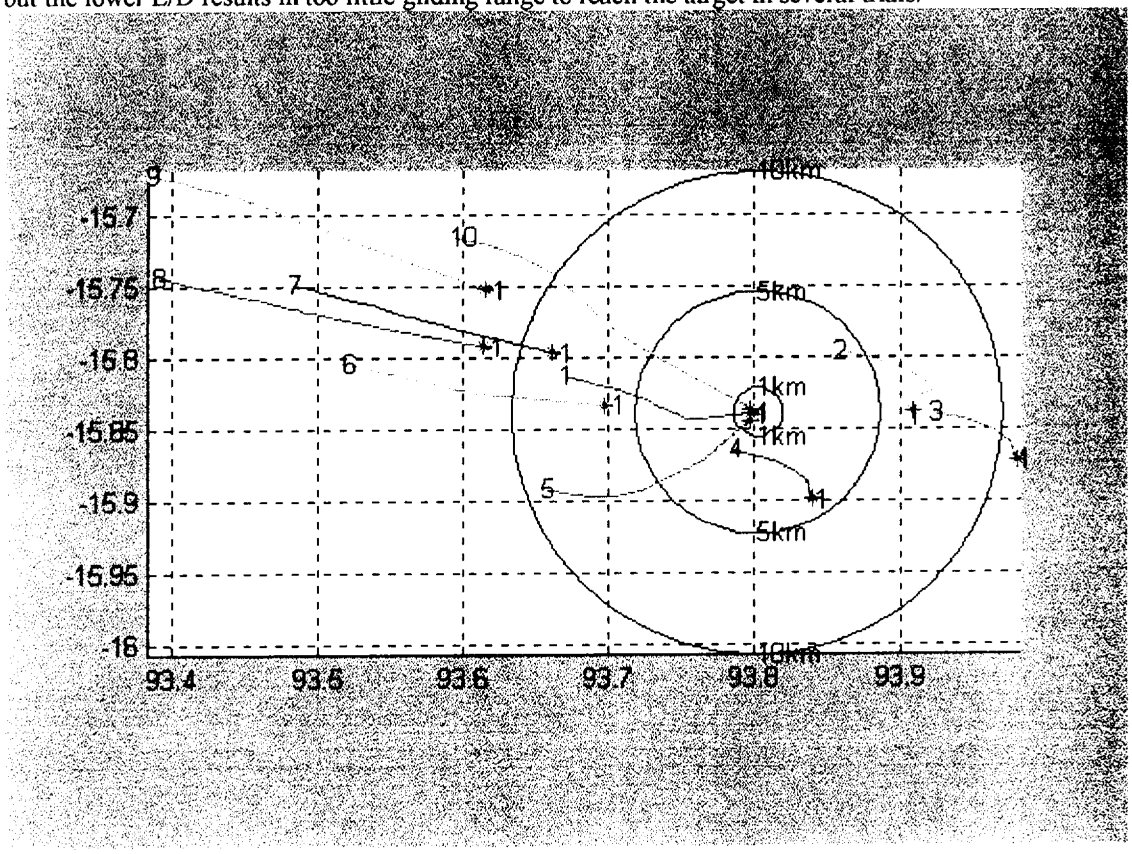
First Guesses.
First guesses for the first 2 scenarios were provided by calculation from within Matlab. The set of initial conditions was known as was the desired reference target. From this data and some brief trigonometry, it is fairly trivial to determine initial guesses for the targeting algorithm within POST. Rather than discuss in detail here, refer to the previous performance report: "An Investigation of Terminal Guidance and Control Techniques for a Robotic Mars Lander, performance report submitted under NASA Grant NAG-1-2086 for the period 4/15/98 to 5/15/99". The particulars are similar.

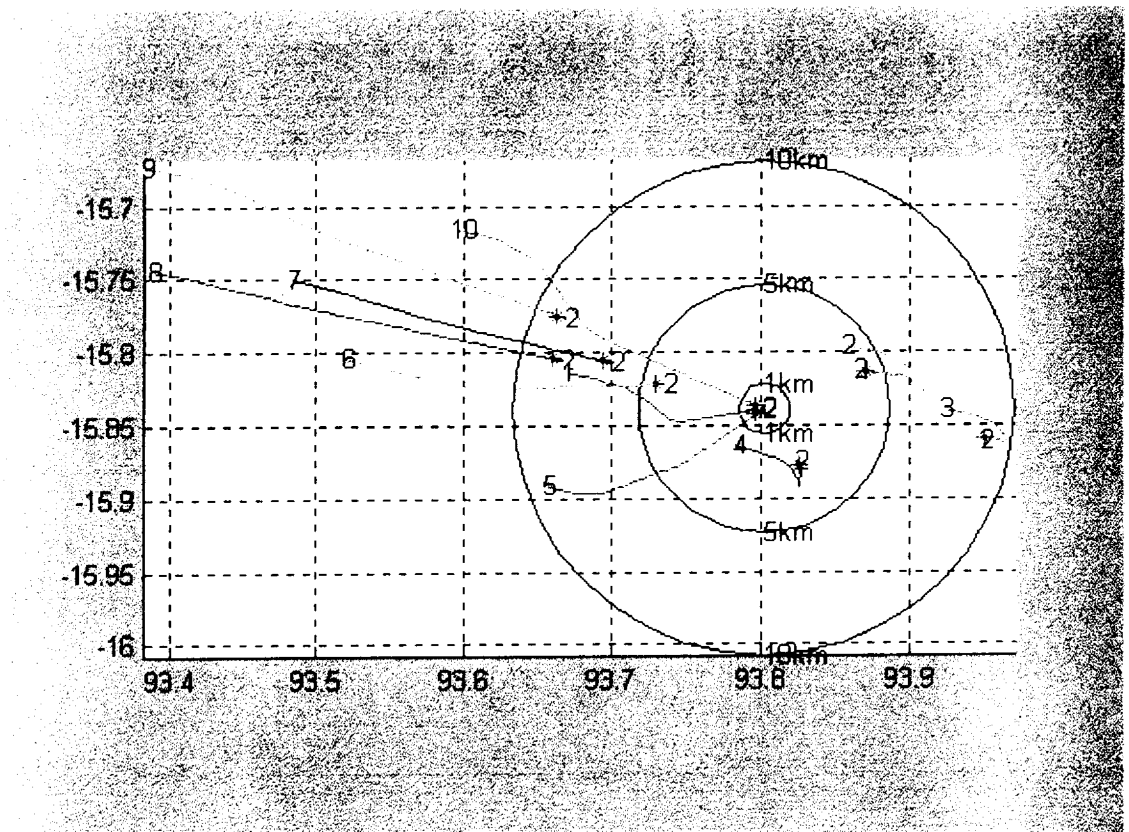
For the 'Hover and Thrust' and 'Lifting Body' scenarios discussed in the attachment, first guesses were manually input for each trial rather than automated as above. However, a similar logic was used for determination of values.

Results.

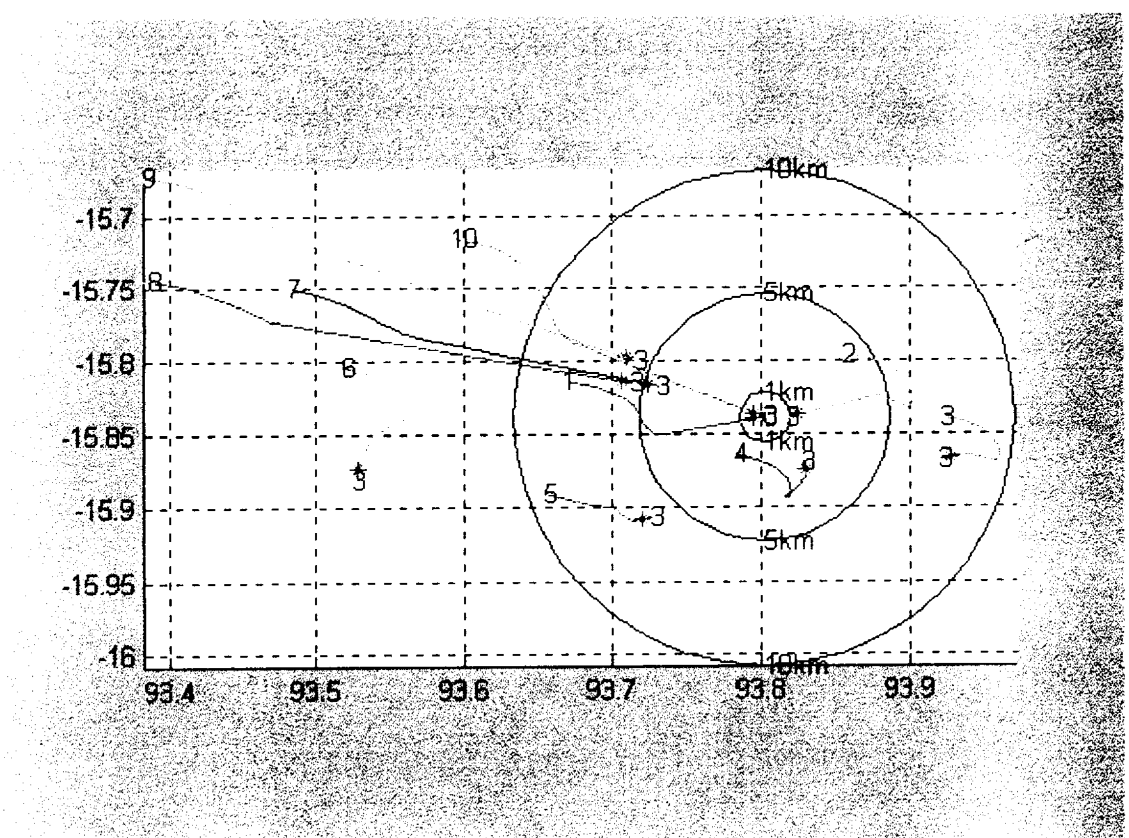
Since the attachment covers most of this area, this will be a brief discussion of what is not covered there. All of the following pictures show 10 initial conditions and their associated trajectories. It is easy to see from straight inspection that the midlevel L/D values do the best at minimizing distance to target. The hover and thrust cases using the methodology described in this paper do not do well mainly because of lack of infinite fuel. The attached paper shows much improved results and is more current.

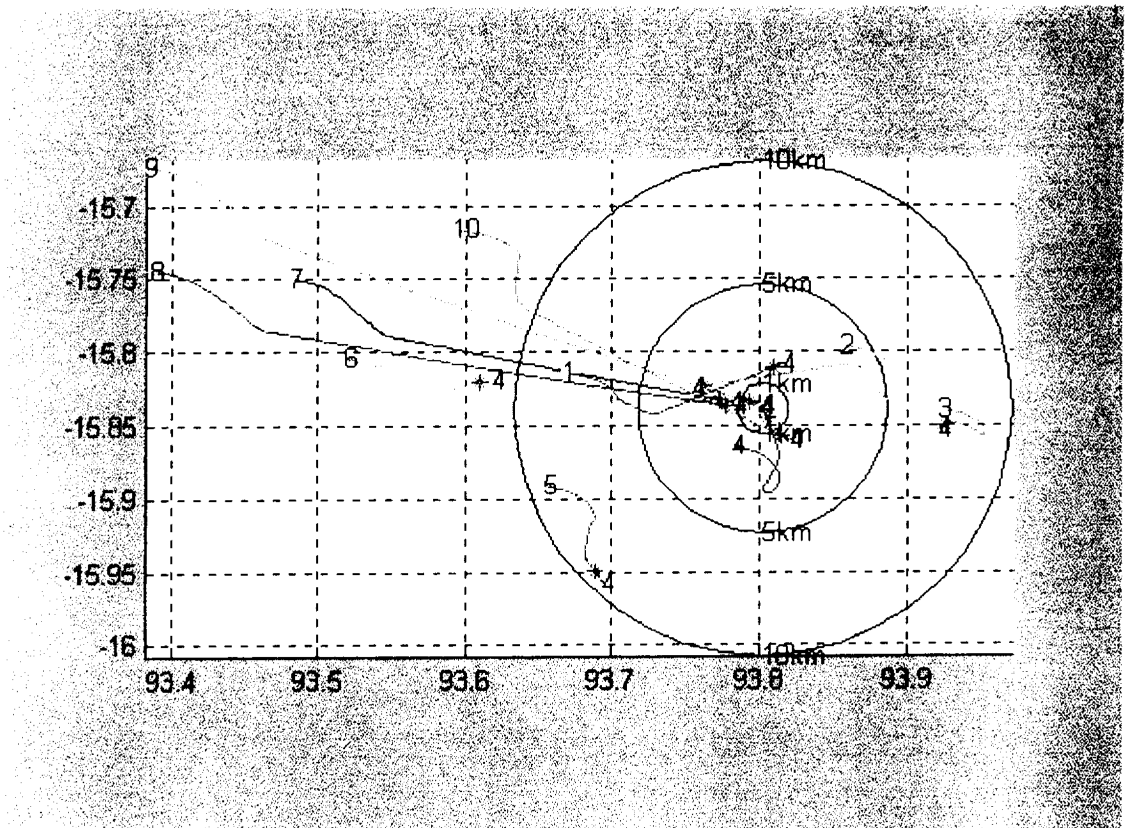
This picture shows the first 10 initial conditions (out of 76) and their associated trajectories for the L/D=0.5 lifting chute case. The numbers on the left of the trajectory indicate which trial (1 is the mean reference) and the numbers on the left (in this case all 1 indicate which L/D case has run (internal to Matlab simulation). Notice that the trajectories look pretty good but the lower L/D results in too little gliding range to reach the target in several trials.



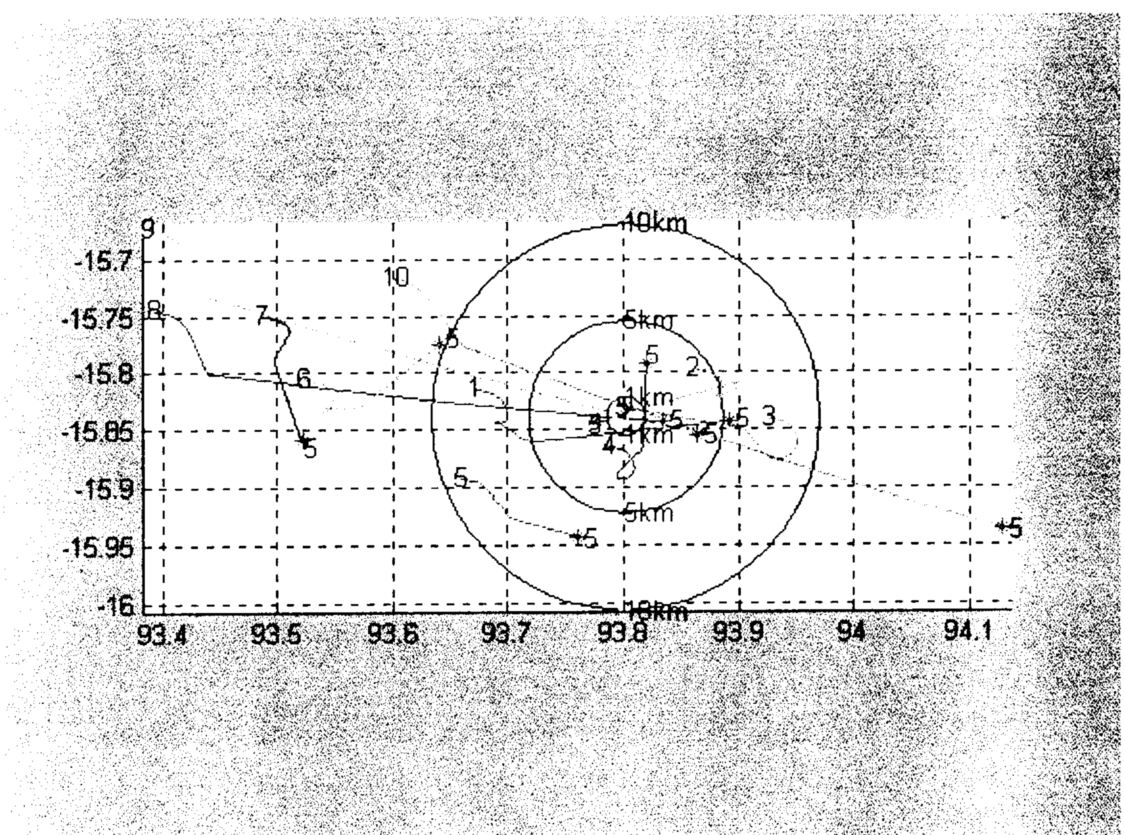


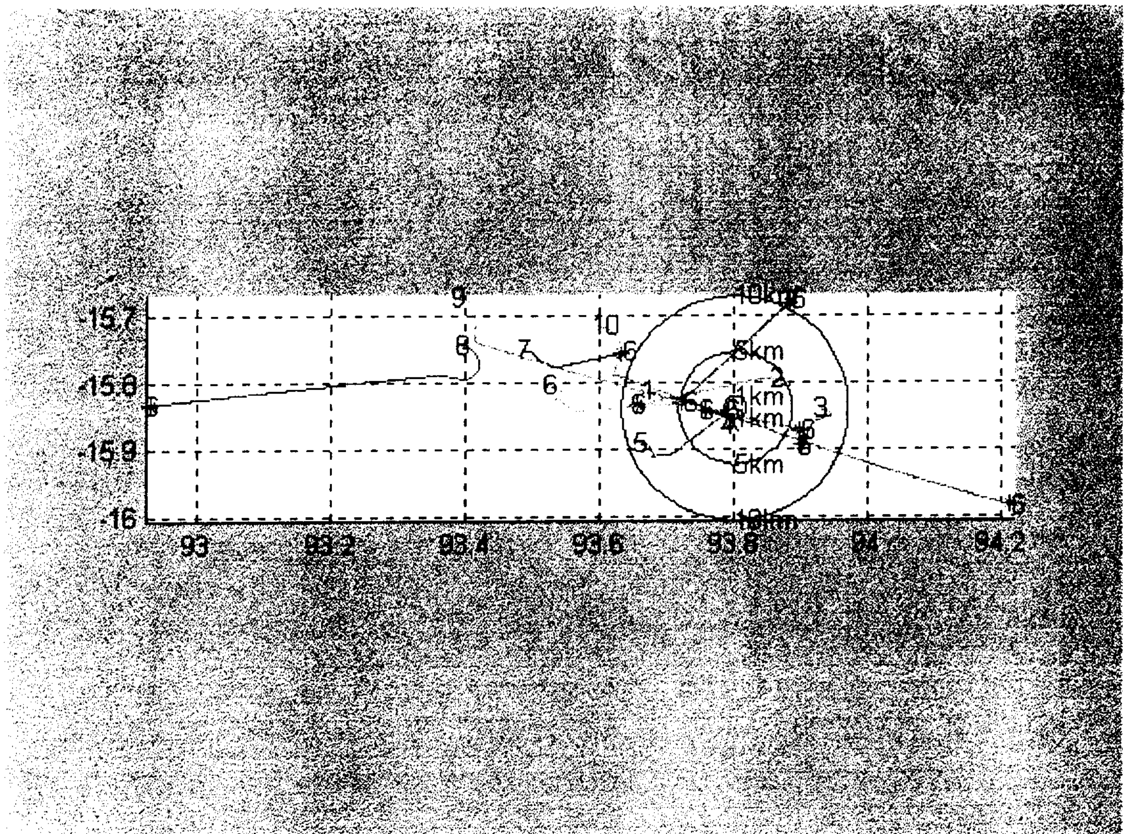
Above is the L/D=0.75 and below L/D=1.0, both with fairly good results. Notice that no matter what the L/D is, the simulation has trouble turning completely around (i.e., overshoot case).





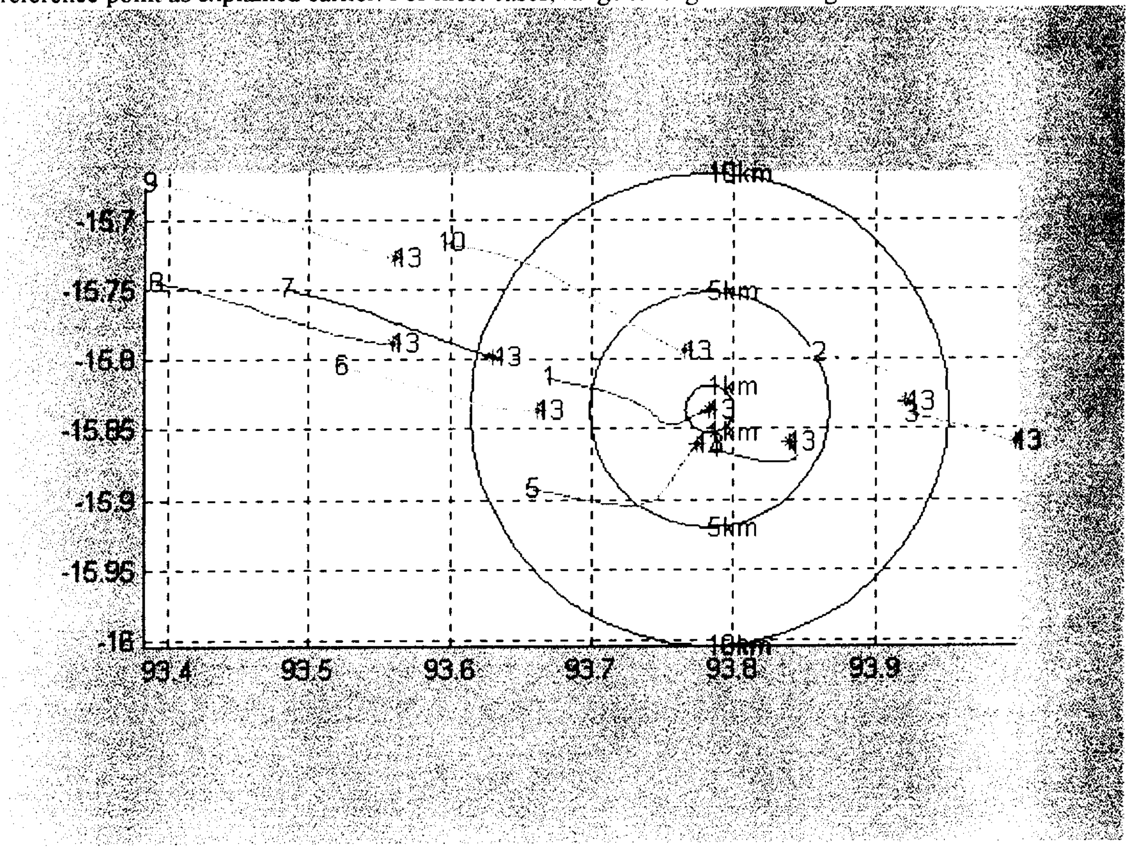
Above L/D=1.5 and below L/D=2.0. Here we start to see the effects of high L/D in range of glide but also in loop maneuvers. During high speed of entry with high L/D, a stallout condition can occur that negates control algorithm use.

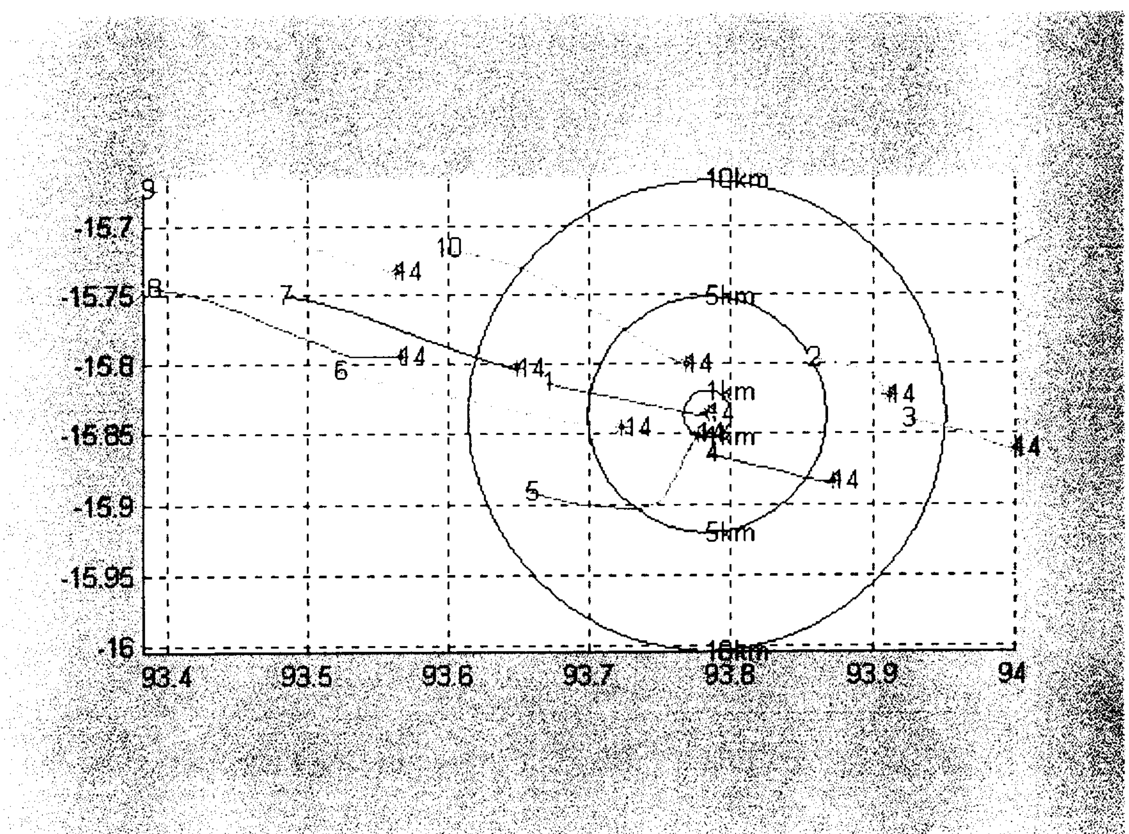




Above is the highest L/D tested, 2.5. As can be seen glide distance is great but at the expense of high number of stallouts that are trouble for this particular control strategy.

Below are the two thrusting on a parachute cases. They are run with the same initial conditions but slightly different reference point as explained earlier. For most cases, range to target is too long for fuel available.





```
1$search
c***********************************************************************
c               m2001newbase2.inp     6/29/99
c
c     Mars Lander from LaRC hand-off to 2500 m
c
c
c     Parachute diameter = 13.0 m
c
c     Marsgram atmosphere - Feb. 3, 2002; 00 hr, 00 min, 0.0 s
c     ln(pres) and atem input as tables
c     Marsgram winds*Braun multipliers input as tables
c
c     No optimization, just targeting to constraints
c
c     Parachute is straight ballistic
c     Only aoa and thrust level are controls during rev grav turn
c
c     This deck is a modified version of EMQ-grvtn-newref.inp
c     Includes most recent M2001 chute handoff conditions & ref (south site)
c     Includes more up to date engine
c     (from CPIA/M5 Liquid propellant engine manual - unit 187):
c           Viking lander(max throttle cond):
c                  Isp=210sec, Thrust=632lbf, exit area=1.588in^2
c                  Flow rate=3.10 lb/s., thrust coeff 1.53
c
c     Last modified 6/28/99
c
c     Constraints:
c      At event 80 (critr = wr = 2.0) -
c      2499 < gdalt < 2501
c      -1.1 < ur    < 1.1
c      -1.1 < vr    < 1.1
c
c***********************************************************************
     ioflag    = 3,       / metric input, metric output
     ipro      = -1,      / trajectory print flag
c***********************************************************************
c Optimization variable - 5.b
c - general optimization inputs
     irscl     = 3,       / default value (flag)
     isens     = 0,       / forward finite differences, default value (flag)
     maxitr    = 50,      / maximum number of iteration
     opt       = 0,/-1,
     optph     = 80,
     optvar    = 'dprngl',
     srchm     = 4,       / projected gradient
     wopt      = 1.0,     / weighting for optimization variable (default val)
c - projected gradient specific inputs, all defaults except coneps(2)
     coneps    = 89.9, 4*1.0e-04,
     consex    = 1.0e-6, 0.001,
     fiterr    = 1.0e-6, 0.001,
     gamax     = 10,
     ideb      = 0,
     npad      = 9.0, 4.0, 14.4494,
     pctcc     = 0.3,
     pdlmax    = 2.0,
     pgeps     = 1.0,
     prntpd    = 0,
     p2min     = 1.0,
     stminp    = 0.1, 0.1,
     stpmax    = 1.0e+10,
c***********************************************************************
c Constraint variable - 5.c
c - general Dependent Variable  inputs
     depph     = 80,80,80,80,80,
```

```
     depvr    = 'gdalt','ur','vr','ur','vr',
     ifdeg    = 0,0,0,0,0,
     indxd    = 1,2,3,4,5,
     ndepv    = 5,
c - projected gradient specific inputs
     deptl    = 1.0,0.1,0.1,0.1,0.1,
     depval   = 2500,1.0,1.0,-1.0,-1.0,
     idepvr   = 0,1,1,-1,-1,      / constraint types
c***********************************************************************
c Controls - 5.d
c - general Independent Control inputs
     indph(1) = 50,50,
c
     indvr(1) = 'etapcl','alppcl',
c
     nindv    = 2,
     pert(1)  = 1.0e-4,1.0e-4,
c
     u(1)     = 1.0,0.0,
c
c - projected gradient specific input
     modew    = 1,
 $
c***********************************************************************
c     Trajectory Simulation Inputs
c***********************************************************************
1$gendat
     title=0h*EMQ-grvtn.inp*,
     event       = 1,          / first event number
     fesn        = 500,        / final event number
c - NUMERICAL INTEGRATION METHODS p. 6.a.15-1
     npc(2) =1,              / integration method (using RK) [flag]
     dltmax =1,              / max step size when using variable steps [s]
     dltmin =0.05,           / min step size  "      "       "       "   [s]
     dt=1,                   / integration time step [s]
     kstpmx=5,               / max # of integration steps for each integration
     npinc=5,                / # of integration steps on each cycle
c***********************************************************************
c  Initial Event Conditions/Setup
c
c - INITIAL POSITION AND VELOCITY p. 6.a.12-1
     npc(3)=1,                    / initial position in xi, yi, zi
     npc(4)=1,                    / initial velocity in vxi, vyi, vzi
     npc(40)=3,                   / reference plane for azimuth and FPA [flag] 3=?
c - ---- ---- ---- ---- ---- ----
c - initial conditions for batch runs, includes 'timeo' (via Matlab)
*include '../../pvstates.dat',
c - ---- ---- ---- ---- ---- ----
c
c - RANGE CALCULATIONS p. 6.a.19-1
     npc(12)=1,/ cross/down range option [flag] (3=?)
    / lonref=93.6484317,/ from Scott (new M2001 south site)
    / latrefgd=-15.8108183,
     lonref=93.8023,/ averaged from lifting chute bank studies
     latrefgd=-15.8384,
c
c - PARACHUTE MODEL p. 6.a.28-1
     npc(32)=2,
     diamp(1)=0.0,/ init val of chute diam, unfurl to 13m
     drgpk(1)=1,
     idrgp(1)=0,
     parif(1)=70.0,
c
c - AERODYNAMIC INPUTS p. 6.a.1-1
     npc(8)=3,/aerodynamic coefficient [flag]
     sref=4.5238934,/aerodynamic reference area [m^2] (from M98)
```

```
c
c - AEROHEATING CALCULATIONS p. 6.a.2-1
     npc(15)=1,/ calculate aeroheating rate & tot. heat using Chapman
     npc(26)=0,/ no special aeroheating calculations
     rn= 0.6638,/ nose radius for Chapman heating (M98nom.inp)
c
c - ATMOSPHERE PARAMETERS p. 6.a.4-1
     npc(5)= 6,/ 2/3/02, 0 hr Marsgram atmosphere input as tables
     npc(6)= 2,/ Marsgram winds* Braun multipliers input as tables
     atmosk(1)=241.0,
     atmosk(2)=5.335e-03,
c
c - CONIC CALCULATION OPTION p. 6.a.5-1
     npc(1)= 3,/Keplerian conic option [flag]
     mre=1hu,/value of mean radius to be used [m](1hu = [re+rp]/2)
c
c - GRAVITY MODEL p. 6.a.10-1
     npc(16)=0,/ spherical or oblate model (oblate) [flag]
     j2=0.1958616e-02,/ spherical harmonics of gravity potential function
     j3=0.3144926e-04,
     j4=-0.1889437e-04,
     j5=0.2669248e-05,
     j6=-0.1340757e-05,
     j7=0.0d0,
     j8=0.0d0,
     re=3393940.0,/ equatorial radius [m]
     rp=3376780.0,/ polar radius [m]
     mu=4.28282868534e+13,/ gravitational constant (mars) [m^3/s^2]
     omega=7.088218e-05,/ rate of rotation of planet [rad/s]
     go=3.718,/ weight to mass factor (Mars surface)
c - VELOCITY LOSSES p. 6.a.25-1
     npc(25)=2,/ velocity loss calculation
c***********************************************************************
c    Initial Guidance Inputs
c***********************************************************************
     iguid(1)=0,0,1,/ atm.rel. aero angle guidance
     alppc(1)=180.0,/ initial alpha
     maxtim=2000.,/ maximum time
     altmax=550000.,/ maximum altitude
     altmin=-3000.0,/ minimum altitude
c***********************************************************************
c  Vehicle Model
c***********************************************************************
     wgtsg=2176.811,/veh. wt. at parachute deploy, N (585.479 kg)
     wpropi=372.0,/initial propellant weight, N (100 kg)
     npc(30)=0,/enhanced (component) weight model
     npc(9)=1,/rocket engine
     npc(27) = 1, / integrate flow rate of specified engines
     npc(22)=2,/input all four coef's in throttling parameter
     neng=2,/2 engines
     ispv(1)=553.9,553.9,/ Mars Isp (Earth Isp = 210 sec)
     iwdf(1)=2,2,/flow rate = vac. thrust/ispv
     iwpf(1)= 0,0,
     iengmf(1)=0,0,/engine off initially
     iengt(1)=0,0,/fixed engine angles (in tables) w.r.t body
c***********************************************************************
c  Print Block
c - PRINT VARIABLE REQUESTS p. 6.a.16-1 --->>
     npvl=0,/ # of print variables per line [flag]
     pinc=10,/ print interval
     prnc=0,/ make profile for plotting
*include '../../prntblk.dat',   / printing variables
 $
l$tblmlt
 vwum = 1.0,
 vwvm = 1.0,
```

```
 $
l$tab
 table = 'denkt',1,'gdalt',3,1,1,1,
 0.0,1.0,30000,1.0,130000,1.0,
 $
l$tab
 table = 'prest',1,'gdalt',6,1,1,1,
 0,6.52553,1859,6.36383,4217,6.15610,6387,5.96265,8515,5.77084,
 8816,5.74359,
 $
l$tab
 table = 'atemt',1,'gdalt',6,1,1,1,
 0,226.1419,1859,223.766,4217,220.502,6387,217.854,8515,215.292,
 8816,214.936,
 $
l$tab
 table = 'vwut',1,'gdalt',6,1,1,1,
 0,-0.1248,1859,0.02672,4217,0.19915,6387,0.55803,8515,0.86006,
 8816,0.85601,
 $
l$tab
 table = 'vwvt',1,'gdalt',6,1,1,1,
 0,0.200574,1859,0.42458,4217,0.68911,6387,1.5575,8515,2.2659,
 8816,2.2508,
 $
l$tab
 table = 'vwwt',0,0.0,
 $
l$tab
 table = 'tvc1t',0,2646.0,
 $
l$tab
 table = 'tvc2t',0,2646.0,
 $
l$tab
 table = 'ae1t',0,0.001,
 $
l$tab
 table = 'ae2t',0,0.001,
 $
l$tab
 table = 'pi1t',0,180.0,    / engine #1 gimbal pitch angle
 $
l$tab
 table = 'yi1t',0,0.0,      / engine #1 gimbal yaw angle
 $
l$tab
 table = 'pi2t',0,0.0,     / engine #2 gimbal pitch angle
 $
l$tab
 table = 'yi2t',0,0.0,   / engine #2 gimbal yaw angle
 $
l$tab
 table = 'wd1t',0,12.98, / engine #1 flow rate
 $
l$tab
 table = 'wd2t',0,12.98,    / engine #2 flow rate
 $
l$tab
 table='cdt',0,1.7,
 $
l$tab
 table='clt',0,0.0,
 $
l$tab
 table='cdp1t',0,0.41,
```

```
 endphs=1,
 $
c
c Parachute fully deployed
l$gendat
 event=22.,1.,
 critr='diampl',
 value=13.0,
 parif(1)=0.0,
 wgtsg = 1937.297,
 endphs=1,
 $
c - - - convert parachute to dragbody for comparison with liftchute case
l$gendat
 event=25.,
 critr='tdurp',
 value=0.0,
 npc(32)=0,       /turn off chute
 sref=132.73   /surf area for chute of 13m, neglect lander surf area
 $
l$tblmlt
 $
l$tab
 table='cdt',0,0.41,  /lander now has drag of chute
 endphs = 1,
 $
c - - - jetison parachute; turn on engine #1 (start of reverse gravity turn)
l$gendat
 event = 50,
 critr = 'gdalt',
 value = 3500.0,
 diamp(1)=0,
 wjett = 276.702,           / weight of parachute lost (N)
 sref=2.0,                  / surface area of lander final configuration
 iengmf(1) = 1,0,
 iwpf(1) = 1,0,
 iguid(1) = 0,      / aero-guidance
 iguid(2) = 1,      / individual component steering
 iguid(6) = 1,      / alpha input by targeting algo
 iguid(7) = 0,      / beta carried over
 iguid(8) = 0,      / bnkang carried over
 $
l$tblmlt
 $
l$tab
 table='cdt',0,2.0,  / drag of lander without chute
 endphs = 1,
 $
c - marks 2500 meter mark (Martian surface)
l$gendat
 event=80,
 critr='wr',
 value = 2.0,
 iengmf(1) =  0,0,
 iwpf(1) = 0,0,
 endphs=1,
 $
c
c*******************************************************************************
c     This event marks arrival at the  Martian surface
c*******************************************************************************
l$gendat
 event=500,
 critr='tdurp',
 value=0,
 endphs=1,
```

```
endjob=1,
endprb=1,
$
```

Space 2000 paper

AIAA 2000-5342

Terminal Guidance Techniques for a Mars Precision Lander

Gerald Walberg*
Walberg Aerospace

Brian K. Birge**
Mars Mission Research Center, North Carolina State University

Abstract

Precision landing for a Mars mission has been investigated. Two descent scenarios were studied: (1) A ballistic parachute and powered gravity turn to hover conditions followed by a constant altitude, powered lateral translation to the target landing site and (2) A guided, lifting parachute followed by a powered gravity turn to the target landing site. Both approaches achieved terminal conditions within 10 m of the landing site. The powered lateral translation scenario, however, required from 72 to 84 kg of additional propellant with consequent payload reductions. The lifting vehicle required a larger parachute (18 m dimeter compared to 13 m for the ballistic parachute) but the propellant savings resulting from the lower velocities at rocket ignition compensated for the larger mass of the lifting chute. No estimates of the mass, size and complexity of the lifting-parachute actuator system are available at this time. However, it seems likely that the additional mass would be significantly less than the mass of propellant needed for the powered lateral translation maneuver.

Nomenclature

azvelr	Azimuth of the relative velocity vector, measured from north
gammar	Relative flight path angle
gdalt	Geodetic altitude
gdlat	Geodetic latitude
long	Longitude
tau	MarsGRAM dust factor
V, velr	Relative velocity

*President, Fellow
**Graduate Research Assistant

Introduction

As the exploration of Mars proceeds there will be an increasing need for a precision landing capability. This will be particularly true for robotic sample return missions and crewed missions. While several successful Mars landings have been carried out, the present state-of-the-art does not allow true precision landings.

The Viking Missions, which featured a lifting entry from Mars orbit followed by a ballistic parachute phase and a rocket powered final descent and touchdown, had a 3-sigma touchdown ellipse semi-major axis of approximately 120 km and achieved estimated touchdown miss distances of 24 km (Viking 1) and 9 km (Viking 2) [1]. The Pathfinder Mission carried out a direct, uncontrolled ballistic entry followed by a ballistic parachute descent and an air-bag landing. It landed 23 km from the center of its 300 by 100 km science requirements ellipse.[2] The ill-fated Mars Polar Lander was designed to carry out an actively controlled zero-lift entry followed by a ballistic parachute phase and a rocket powered final descent and touchdown. It had a 3-sigma touchdown ellipse of 200 by 20 km. As described in Ref. 2, the MSP '01 lander would have carried out an actively controlled and guided lifting entry followed by a ballistic parachute phase and a rocket powered terminal descent and touchdown. Monte Carlo analyses showed this vehicle to have a 98.5% probability of touching down within 5 km of its target-landing site. [2]

If greater landing precision is to be obtained, an actively controlled and guided terminal phase with significant lateral maneuvering capability must be employed. This paper presents results obtained from an investigation of two terminal-phase guidance techniques for a Mars lander similar to that described in Ref. 2 for the Mars Surveyor 2001 Mission. Special emphasis is placed on the use of a lifting, guided parachute to provide lateral ranging capability.

Lander Characteristics

The Mars lander analyzed in this paper is essentially the same as that considered in Ref. 2, i.e. it is based in large part on the ill-fated Mars Polar Lander. The NASA Mars Program is currently being restructured and future lander designs may differ considerably from the family of vehicles that was studied in detail prior to the loss of the Polar Lander. The present investigation is, however, intended to be complementary to Ref. 2. Hence, a similar lander is analyzed.

At entry into the Martian atmosphere, the lander is housed within a 2.65 m diameter 70° sphere-cone aeroshell similar to the Viking configuration. The entry vehicle uses aeromaneuvering to deliver the lander to an acceptable hand-off condition having a dynamic pressure between 400 and 1175 N/m^2, a Mach number between 1.60 and 2.28 and an altitude between 6.5 and 17.0 km. The aft aeroshell is then separated and, for the baseline lander, a 13 m diameter ballistic, disk-gap-band parachute is deployed and decelerates the lander to approximately 80 m/s. At 3.5 km altititude, the parachute is staged off and a hydrazine rocket engine is ignited and provides gravity-turn deceleration to a touchdown velocity of approximately 2 m/s. The nominal landed mass is 374 kg.

The lifting-parachute version of the lander employs a larger 18 m diameter parachute. This parachute is deployed initially as a conventional disk-gap-band chute and retains this configuration (with its' demonstrated supersonic deployment capability[3]) until the vehicle Mach number is less than 0.8. The parachute is them converted to a lifting configuration by using burn wires to cut away two panels in the ring to create drive slots similar to those used in smoke jumper parachutes. The resulting lifting parachute is steerable, has an L/D of 0.8 and an equilibrium glide slope of 51 deg. .

This lifting parachute configuration is presented schematically in Fig. 1 and the lander characteristics are presented in Table 1.

Trajectory Simulations

Algorithm

The 3-D version of the Program to Optimize Simulated Trajectories (POST)[4] was used analyze the lander. In this algorithm, as it is utilized in the present investigation, the point-mass trajectory equations are solved using 5th-order Runge-Kutta integration. Mars is modeled as an oblate planet with a gravity field described by an eight-term spherical harmonic potential function. POST has both targeting and optimization capability. In the present study, the projected gradient optimization algorithm was used.

Initial Conditions

The initial conditions for the present trajectory analyses were taken from the hand-off conditions presented in Ref. 2 for an entry vehicle having an L/D of 0.12. In Ref. 2, a two-thousand-case Monte Carlo analysis was carried out to statistically assess the performance of the entry vehicle in the presence of atmospheric, aerodynamic, mass property, control system, inertial measurement unit, and entry state uncertainty distributions. In the present study, eleven cases representing the outer bounds of the hand-off ellipse presented in Ref. 2, are analyzed. These hand-off (initial) conditions are presented in Table 2.

Mars Atmosphere

The Martian atmosphere was described by the Mars Global Reference Atmosphere Model[5] version 3.7. From Table 2, it is seen that each case has a different tau associated with it. Tau is a factor, used in the present version of MarsGRAM, that indicates the degree of dustiness of the Martian atmosphere. The present analyses assume, as did those of Ref. 2, that all lander descents occur on February 3, 2002. This is late in the dust storm season on Mars and as a result, there is the possibility of significant variations in atmospheric density. The Monte Carlo analysis of Ref. 2 assumed a tau variation from 0.3 to 1.6. As can be seen from Table 2, the hand-off cases considered in the present investigation also cover a wide range of tau from 0.438 to 1.55. Atmospheric density profiles for the maximum- and minimum-tau cases are presented in Fig. 2. Note that the density variation is approximately 33 percent.

Descent Scenarios

Two descent scenarios were investigated: (1) Ballistic parachute followed by a powered gravity turn to hover conditions followed by a constant altitude, powered lateral translation to the target landing site and (2) Guided, lifting parachute followed by a powered gravity turn to the target landing site.

The initial conditions are presented in Table 2. The terminal constraints are: (1) geodetic altitude = 2500 m (to account for possibly mountainous terrain) and (2) all three components of relative velocity < 2.0 m/s.

The parachutes are modeled as being reefed with deployment times of approximately 2 and 3 seconds for the ballistic and lifting parachutes respectively. As mentioned previously, the lifting parachute maintains its' initial ballistic configuration until the Mach number is < 0.8.

The following controls are used to satisfy the terminal constraints:

1. – Rocket engine thrust level and throttling – modeled as a 2nd order polynomial in time.
2. – Lander angle-of-attack and sideslip during powered flight – modeled as 2nd order polynomials in time.
3. – Lander bank angle during lifting parachute descent – assumed constant.
4. – Time after M = 0.8 when the drive slots of the lifting parachute are opened.

Results

In Fig. 3, the latitude and longitude coordinates of the hand-off points (initial conditions) and the points where the ballistic parachute lander reached hover conditions are presented. Also shown is the target landing site. In table 3, the mass of the lander at the hover condition, the distance from the landing site (miss distance) and the propellant usage up to this point are shown. Note that the landing site is near the middle of the hover ellipse, miss distances at hover range from 3.6 to 5.0 km and the propellant usage is modest (34.2 to 42.1 kg). The hover ellipse is somewhat smaller than the hand-off ellipse. This is the result of optimizing the angles of attack and sideslip during the powered phase of the descent. The miss distances shown in Table 3 are the smallest that could be achieved (while meeting the terminal constraints) without a dedicated lateral translation maneuver. When the additional lateral translation maneuver is performed, significant additional propellant is required, resulting in the landed masses and propellant usage shown in Table 4. The lateral translation maneuvers bring the lander quite close to the landing site (within 8.9 m) but require from 72 to 84 kg of additional propellant and hence would have payloads that were reduced by this amount.

In order to achieve comparable landing precision without these large payload reductions, the lifting, guided parachute was investigated. Descent histories for the lifting-parachute lander (case 4) are presented in Fig.4. Landed masses and miss distances for all eleven cases are presented in Table 5.

Case 4 is illustrated because it is the most difficult of the lifting chute cases. It has the lowest hand-off altitude and the lowest atmospheric density, and significant lateral maneuvering is required. As shown in Fig. 4(a), the parachute is converted to the lifting configuration approximately 8 sec. after parachute deployment. This is very soon after reaching M = 0.8 and provides maximum benefit from the lifting chute. For some cases – those with high hand-off altitudes and high atmospheric densities and where little lateral maneuvering was required – the ballistic parachute configuration was retained for as long as 67 sec. After reaching M = 0.8. Shortly after conversion to the lifting configuration, the vehicle is banked at – 80 deg. for 17 sec. This aligns the vehicle heading with the landing site as shown in Fig 4(c). From 25 to 97 sec., the vehicle glides, with zero bank angle, toward the landing site. From 97 to 99 sec., a final vernier bank maneuver is performed. The parachute is staged off and the rocket engine is ignited at 3.5 km altitude as shown in Fig. 4(b). From 107.5 to 131 sec. the thrust level, angle-of-attack and sideslip angle are modulated to achieve the required terminal conditions (all three components of relative velocity < 2.0 m/s at 2.5 km altitude). Case 4 has hand-off coordinates of latitude: 13.67 deg, longitude: 269.12 deg and a hand off azimuth of 108.53 deg (see Table 2). Hence, at handoff, the lander is south and west of the landing site, heading east-southeast and must turn through approximately 90 deg. to reach the heading of the landing site. As seen in Fig.3, the various cases have handoff locations north and south of the landing site. Some are within 3.6 km of the landing site. Some are over 13.7 km away. Hence, a wide variety of maneuvers are required of the lifting parachute lander. In some cases, the lander glides past the landing site, turns through nearly 180 deg and glides back to the site.

For all eleven cases, the lifting-chute lander reaches a terminal condition within 9.3 m of the landing site and actually uses less propellant than the corresponding ballistic-chute hover case. The reduced propellant usage is due to the larger diameter of the lifting parachute, which produces a lower velocity at rocket ignition and hence requires less propellant to decelerate to terminal conditions. Note from Table 1 that the lifting chute has a system mass of 20 kg whereas the ballistic chute has a system mass of 9 kg. These system masses were calculated using the equations presented in Ref. 6. The savings in propellant usage nearly compensate for the added mass of the larger chute.

This investigation has not addressed the complexity or the additional mass of the control and actuator system required for the lifting-parachute lander. If, as in Ref.2, the entry vehicle employs aeromaneuvering to reduce the size of the hand-off ellipse, a computer with speed and storage adequate to accommodate the lifting-parachute guidance and control algorithm would already be on board. The most significant additional system would be the actuator system for the lifting chute. No estimates of the mass , size and complexity of this system are available at this time. However, it seems likely that the additional mass would be significantly less than the mass of propellant needed for the powered lateral translation maneuver.

Concluding Remarks

Precision landing for a Mars mission has been investigated. Two descent scenarios were studied: (1) A ballistic parachute and powered gravity turn to hover conditions followed by a constant altitude, powered lateral translation to the target landing site and (2) A guided, lifting parachute followed by a powered gravity turn to the target landing site. Both approaches achieved terminal conditions within 10 m of the landing site. The powered lateral translation scenario, however, required from 72 to 84 kg of additional propellant with consequent payload reductions of this amount. The lifting vehicle required a larger parachute (18 m diameter compared to 13 m for the ballistic parachute) but the propellant savings resulting from the lower velocities at rocket ignition compensated for the larger mass of the lifting chute. No estimates of the mass, size and complexity of the lifting-parachute actuator system are available at this time. However, it seems likely that the additional mass would be significantly less than the mass of propellant needed for the powered lateral translation maneuver.

Acknowledgements

This research was supported by NASA Langley Research Center Grant NAG-1-2086. The authors would like to thank Eric M. Queen for his valuable inputs as technical monitor and Richard W. Powell for providing the parachute hand-off conditions.

References

1. Ingolby, R. N., "Guidance and Control System Design of the Viking Planetary Lander", J. Guidance and Control, Vol. 1, No. 3, May-June, 1978.
2. Braun, R. D., et al, "The Mars Surveyor 2001 Lander : A First Step Toward Precision Landing, IAF-98-Q.3.03, 49th International Astronautical Congress, Sept 28-Oct 2, 1998, Melbourne, Australia.
3. Murrow, H. N. and McFall, J. C. Jr., "Summary of Experimental Results Obtained From the NASA Planetary Entry Parachute Program", AIAA Paper 68-934, 1968.
4. Braur, G. L., Cornick, D. E. and Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)", NASA CR-2770, Feb, 1977.
5. Justus, C. G., "Mars Global Reference Atmosphere Model for Mission Planning and Analysis", J. of Spacecraft and Rockets, Vol. 28, No. 2, pp216-221, April-June 1991.
6. Carter, P. and Smith, R., "Mars Rover Sample Return – Lander Performance", AIAA 89-0633, 27th Aerospace Sciences Meeting, Jan 9-12, 1989, Reno, Nevada.

Table 1 Lander Characteristics

Mass at hand-off conditions	687 kg
Mass at parachute deployment	507 kg
Mass after parachute jettison	
Ballistic parachute	416 kg
Lifting parachute	405 kg
Propulsion system	
Specific impulse	210 sec
Nominal thrust	
Ballistic parachute	2800 N
Lifting parachute	2000 N
Ballistic parachute	
Diameter	13 m
L/D	0
System mass	9 kg
Lifting parachute	
Diameter	18 m
L/D	0.8
System mass (excluding G&C system)	20 kg

Table 2 Hand-Off Conditions

Case	gdlat (deg)	long (deg)	gdalt (km)
1	13.7777	269.0223	1570.13
2	13.7250	269.0468	1519.23
3	13.6921	269.0661	1375.52
4	13.6715	269.1231	9685.22
5	13.6604	269.1712	1242.07
6	13.7250	269.1952	1030.68
7	13.7561	269.2280	1361.51
8	13.7795	269.2048	1402.58
9	13.8004	269.1593	1299.69
10	13.8221	269.1100	1583.00
11	13.8082	269.0738	1584.77

Case	velr (m/s)	gammar (deg)	azvelr (deg)	tau
1	503.02	-17.52	108.29	0.795
2	504.80	-19.08	108.28	0.911
3	504.99	-18.08	108.19	0.538
4	506.21	-21.08	108.53	0.438
5	373.01	-29.73	111.21	1.550
6	506.56	-22.66	108.58	0.612
7	504.39	-25.44	108.65	1.260
8	501.00	-22.81	108.60	1.390
9	501.17	-22.02	108.60	1.090
10	503.87	-16.62	108.38	1.500
11	506.01	-15.57	108.24	0.861

Table 3 Lander Performance: Ballistic Parachute

Case	Landed mass (kg)	Miss distance (km)	Propellant usage (kg)
1	376.1	4.906	39.9
2	375.7	4.021	40.4
3	376.7	4.249	39.4
4	376.9	4.166	39.2
5	380.1	4.398	36.0
6	381.9	4.251	34.2
7	377.7	4.990	38.3
8	381.0	4.416	35.1
9	381.3	3.659	34.8
10	374.0	4.344	42.1
11	375.9	3.555	40.2

Table 4 Lander Performance: Ballistic Parachute Plus Lateral Translation

Case	Landed mass (kg)	Miss distance (km)	Propellant usage (kg)
1	298.1	0.0012	118.0
2	299.9	0.0004	116.2
3	299.5	0.0017	116.6
4	304.4	0.0077	111.7
5	299.4	0.0005	116.7
6	303.3	0.0089	112.8
7	293.6	0.0033	122.5
8	300.1	0.0024	116.0
9	306.8	0.0010	109.3
10	295.1	0.0077	121.0
11	303.7	0.0001	112.4

Table 5 Lander Performance: Lifting Parachute

Case	Landed mass (kg)	Miss distance (km)	Propellant usage (kg)
1	379.1	0.0009	26.2
2	377.6	0.0023	27.5
3	378.9	0.0077	26.2
4	378.4	0.0078	26.8
5	376.0	0.0057	29.1
6	376.7	0.0087	28.4
7	371.6	0.0050	33.5
8	370.1	0.0051	35.0
9	377.2	0.0071	27.9
10	373.6	0.0047	31.5
11	375.3	0.0093	29.9

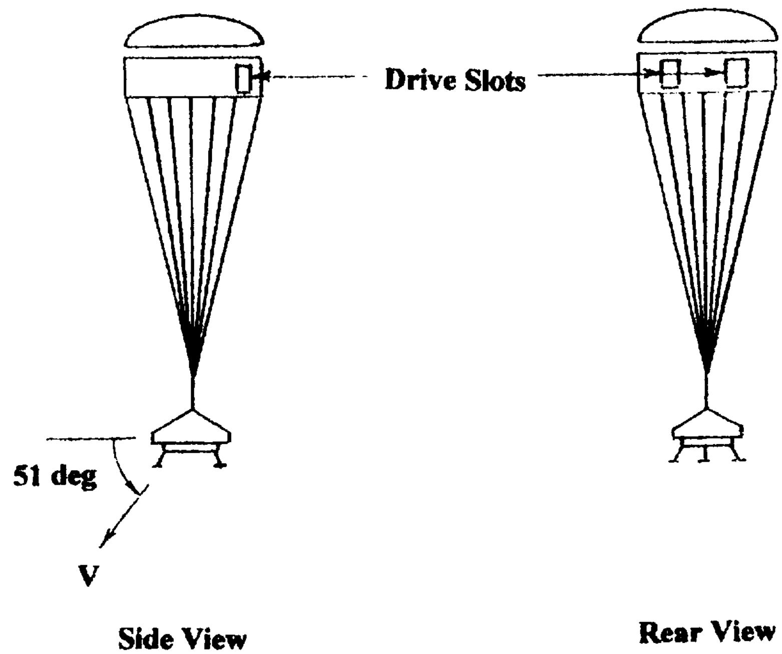


Fig 1. Lifting parachute

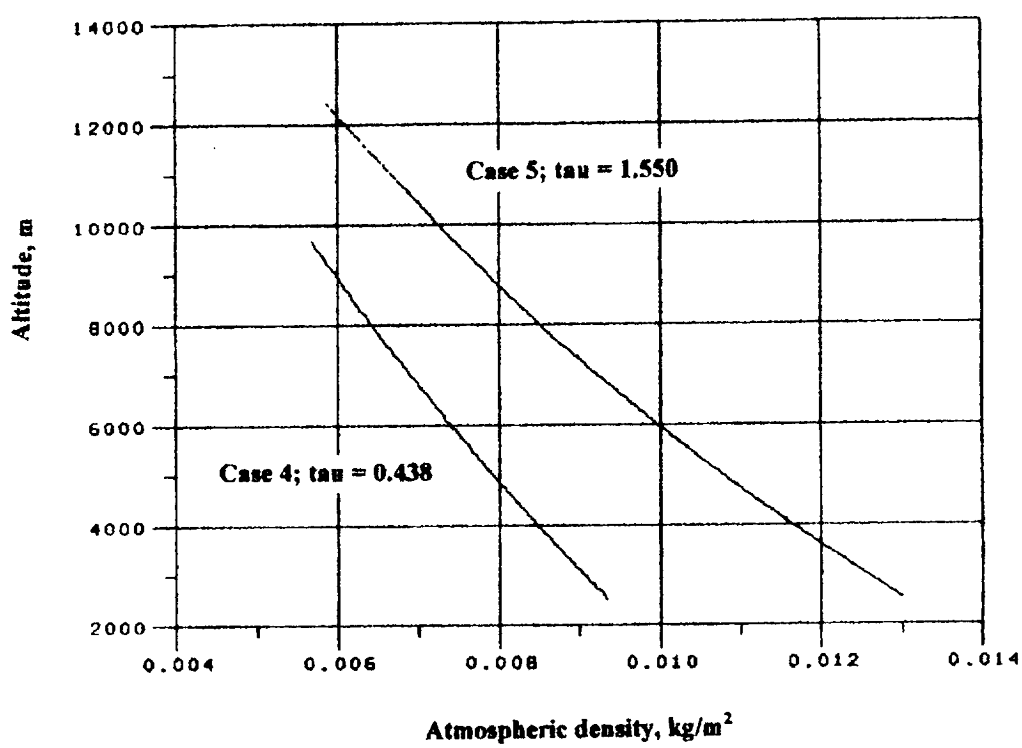


Fig. 2 Atmospheric density profiles

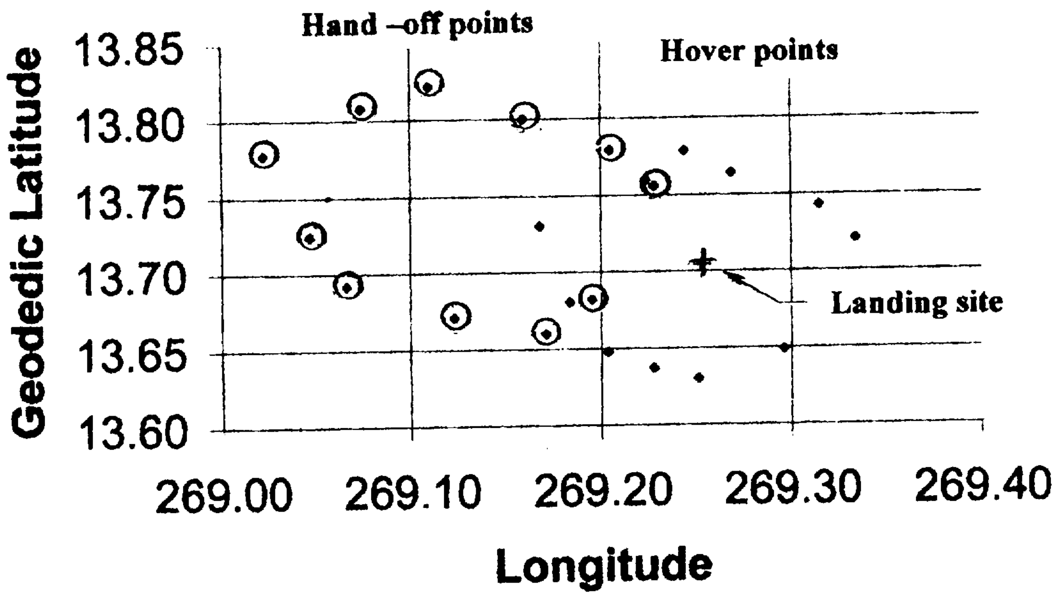
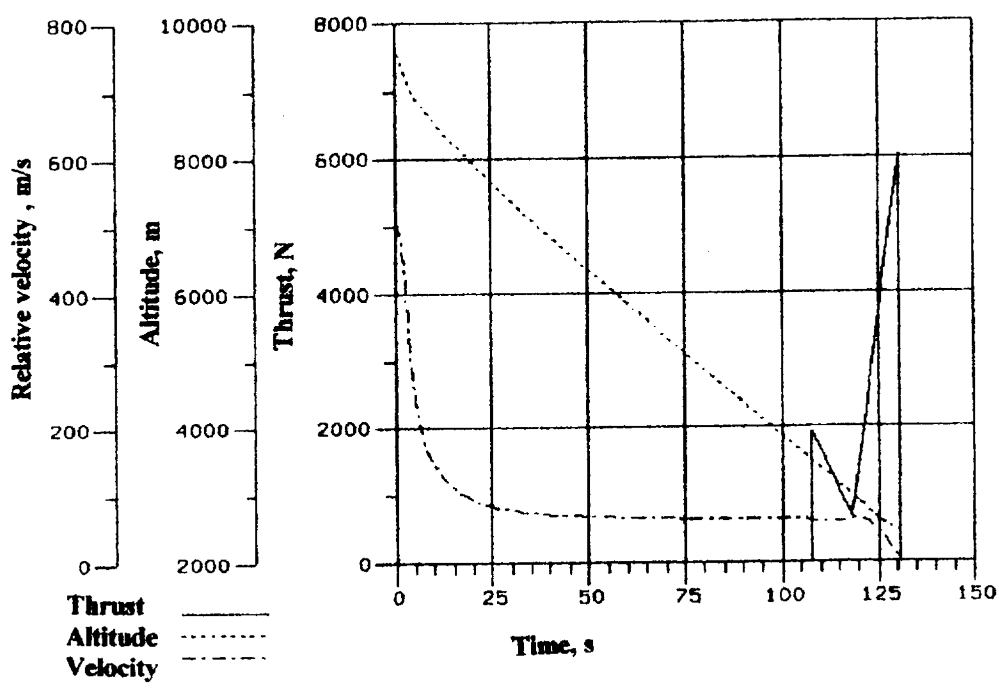
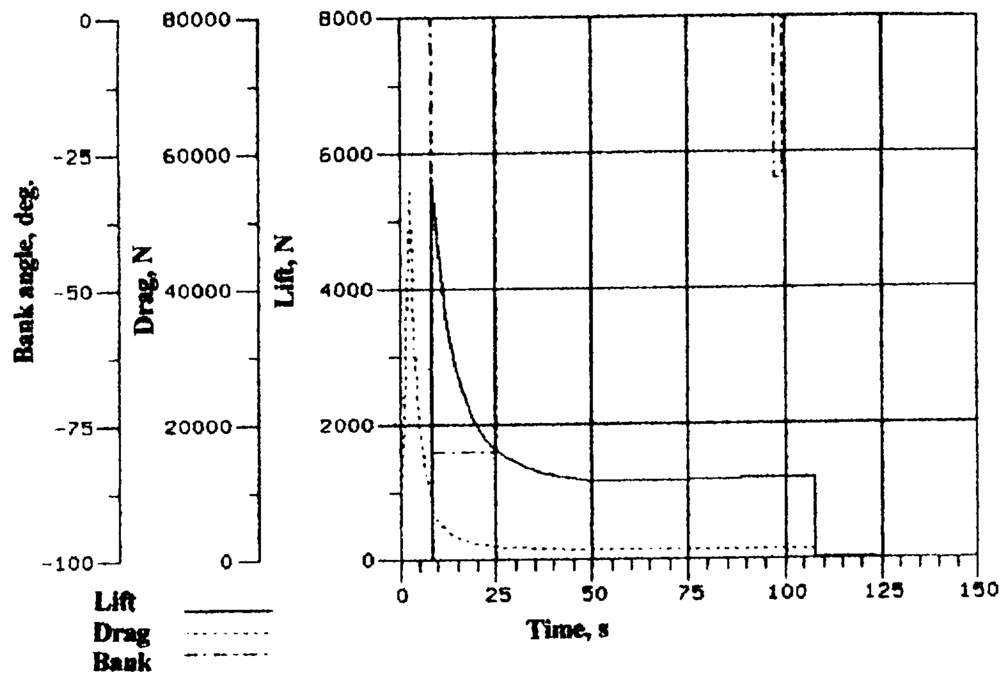


Fig. 3 Hand-off ellipse and hover points for the ballistic parachute



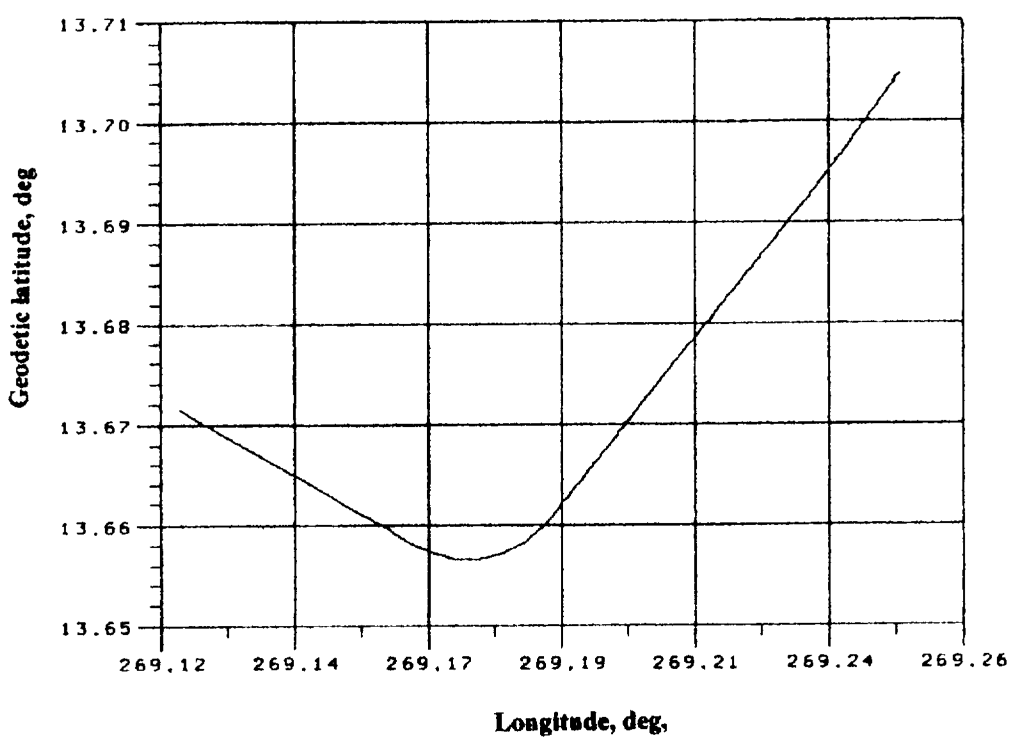
(a) Velocity, altitude and thrust

Fig. 4 Descent histories for case 4 - lifting parachute



(b) Lift, drag and bank angle

Fig.4 – continued



(c) Ground track

Fig. 4 – concluded

```
1$search
c*******************************************************************
c
c              bc
c
c     ballistic parachute
c
c     Mars Lander from RWP parachute deploy conditions to 2500 m
c             Case 17
c
c     Parachute deployed at  velr = approx. 501.8 m/s, deployment time = 1.9 sec
c     Parachute diameter = 13.0 m; Lifting chute, L/D = 1.0
c     Tvac = 2800 N
c
c     Marsgram atmosphere - Feb. 3, 2002; 00 hr, 00 min, 0.0 s
c     Marsgram winds
c
c
c     Constraints:
c    At event 80 (critr = wr = 2.0) -
c      gdalt = 2500;  ur = 2.0;  vr = 20.
c*******************************************************************
     ioflag    = 3,       / metric input, metric output
     ipro      = -1,      / trajectory print flag
c*******************************************************************
c Optimization variable - 5.b
c - general optimization inputs
     irscl    = 3,        / default value (flag)
     isens    = 0,        / forward finite differences, default value (flag)
     maxitr   = 20,       / maximum number of iteration
     opt      = -1,
     optph    = 80,
     optvar   = 'dprng1',
     srchm    = 4,        / projected gradient
     wopt     = 1.0,      / weighting for optimization variable (default val)
c - projected gradient specific inputs, all defaults except coneps(2)
     coneps   = 89.9, 4*1.0e-06,
     consex   = 1.0e-6, 0.001,
     fiterr   = 1.0e-6, 0.001,
     gamax    = 10,
     ideb     = 0,
     npad     = 9.0, 4.0, 14.4494,
     pctcc    = 0.3,
     pdlmax   = 2.0,
     pgeps    = 1.0,
     prntpd   = 0,
     p2min    = 1.0,
     stminp   = 0.1, 0.1,
     stpmax   = 1.0e+10,
c*******************************************************************
c Constraint variable - 5.c
c - general Dependent Variable  inputs
     depph    = 80,80,80,
     depvr    = 'gdalt','ur','vr',
     ifdeg    = 0,0,0,
     indxd    = 1,2,3,
     ndepv    = 3,
c - projected gradient specific inputs
     deptl    = 10.0,0.1,0.1,
     depval   = 2500,2.0,2.0,
     idepvr   = 0,1,1,      / constraint types
c*******************************************************************
c Controls - 5.d
c - general Independent Control inputs
     indph    = 50,50,50,50,
                70,70,70,
```

```
     indvr    = 'etapc1','etapc2','alppc1','betpc1',
                'etapc2','alppc1','betpc1',
     indxi    = 1,2,3,4,5,6,7,
     nindv    = 7,
     pert     = 1.0e-6,1.0e-7,1.0e-5,1.0e-5,
                1.0e-7,1.0e-5,1.0e-5,
     u        =1.0,0.0,90.0,0.0,
                0.0,90.0,0.0,
c - projected gradient specific input
     modew    = 1,
 $
c*********************************************************************
c     Trajectory Simulation Inputs
c**********************************************************************
l$gendat
     title=0h*Mars Lander with lifting parachute *,
     event       = 1,          / first event number
     fesn        = 500,        / final event numbe
c - NUMERICAL INTEGRATION METHODS p. 6.a.15-1
     npc(2) =1,  / integration method (using RK) [flag]
     dltmax =1,  / max step size when using variable steps [s]
     dltmin =0.05,  / min step size  "      "      "       "   [s]
     dt=1,/ integration time step [s]
     kstpmx=5,/ max # of integration steps for each integration
     npinc=5,/ # of integration steps on each cycle
c*********************************************************************
c  Initial Event Conditions/Setup
c
c - INITIAL POSITION AND VELOCITY p. 6.a.12-1
     npc(3)=2,/ initial velocity in planet relative components
     npc(4)=2,/ initial position in planetocentric conponents
     npc(40)=0,/ reference plane for azimuth and FPA [flag] 3=?
     veli = 722.5183,
     gammai = -12.10055,
     azveli = 102.30149,
     gdalt = 15701.326,
     gdlat = 13.77769,
     long = 269.02228,
c
c - RANGE CALCULATIONS p. 6.a.19-1
     npc(12)=3,/ cross/down range option [flag] (3=?)
     lonref=269.1423,/ New target
     latrefgd=13.7363,
c
c - PARACHUTE MODEL p. 6.a.28-1
     npc(32)=2,
     diamp(1)=0.1,/ init val of chute diam, unfurl to 13m
     drgpk(1)=1,
     idrgp(1)=0,
     parif(1)=70.0,
c
c - AERODYNAMIC INPUTS (Entry Vehicle) p. 6.a.1-1
     npc(8)=3,/aerodynamic coefficient [flag]
c     lref=2.4,
     sref=5.515,/aerodynamic reference area [m^2] (from MSP'01)
c
c
c - AEROHEATING CALCULATIONS p. 6.a.2-1
     npc(15)=1,/ calculate aeroheating rate & tot. heat using Chapman
     npc(26)=0,/ no special aeroheating calculations
     rn= 0.6638,/ nose radius for Chapman heating (M98nom.inp)
c
c - ATMOSPHERE PARAMETERS p. 6.a.4-1
     npc(5)= 9,/ 2/3/02, 0 hr Marsgram atmosphere
c     npc(6)= 2,/ Marsgram winds* Braun multipliers input as tables
c     atmosk(1)=241.0,
```

```
c     atmosk(2)=5.335e-03,
c
c - CONIC CALCULATION OPTION p. 6.a.5-1
     npc(1)= 3,/Keplerian conic option [flag]
     mre=1hu,/value of mean radius to be used [m](1hu = [re+rp]/2)
c
c - GRAVITY MODEL p. 6.a.10-1
     npc(16)=0,/ spherical or oblate model (oblate) [flag]
     j2=0.1958616e-02,/ spherical harmonics of gravity potential function
     j3=0.3144926e-04,
     j4=-0.1889437e-04,
     j5=0.2669248e-05,
     j6=-0.1340757e-05,
     j7=0.0d0,
     j8=0.0d0,
     re=3393940.0,/ equatorial radius [m]
     rp=3376780.0,/ polar radius [m]
     mu=4.28282868534e+13,/ gravitational constant (mars) [m^3/s^2]
     omega=7.088218e-05,/ rate of rotation of planet [rad/s]
     go=3.718,/ weight to mass factor (Mars surface)
c - VELOCITY LOSSES p. 6.a.25-1
     npc(25)=2,/ velocity loss calculation
c*******************************************************************
c    Initial Guidance Inputs
c********************************************************************
     iguid(1)=0,0,1,/ atm.rel. aero angle guidance
     alppc(1)=90.0,/ initial alpha
     maxtim=2000.,/ maximum time
     altmax=550000.,/ maximum altitude
     altmin=-6000.0,/ minimum altitude
c********************************************************************
c  Vehicle Model
c********************************************************************
     wgtsg=2554.27,/veh. wt. at parachute deploy, N (585.479 kg)
     wpropi=180.32,/initial propellant weight, N (100 kg)
     npc(30)=0,/enhanced (component) weight model
     npc(9)=1,/rocket engine
     npc(27) = 1, / integrate flow rate of specified engines
     npc(22) = 1, / etapcl = current value
     neng=2,/2 engines
     ispv(1)=553.9,553.9,/ Mars Isp (Earth Isp = 210 sec)
     iwdf(1)=2,2,/flow rate = vac. thrust/ispv
     iwpf(1)= 0,0,
     iengmf(1)=0,0,/engine off initially
     iengt(1)=0,0,/fixed engine angles (in tables) w.r.t body
c********************************************************************
c  Print Block
c - PRINT VARIABLE REQUESTS p. 6.a.16-1 --->>
c     npvl=0,/ # of print variables per line [flag]
     pinc=10,/ print interval
     prnc=0,/ make profile for plotting
     prnt(91) = 4hcdp1,6hdiamp1,6hdiarp1,6hdragpt,6hdragp1,
              2huw,2hvw,2hww,2hur,2hvr,2hwr,4hmach,4hdynp,
              6hspecv1,6hspecv2,6hspecv3,6hspecv4,6hspecv5,
              5hpstop,
 $
l$tblmlt
c vwum = 1.0,
c vwvm = 1.0,
 $
cl$tab
c table = 'denkt',1,'gdalt',7,1,1,1,
c 0.0,0.85,15000,1.25,30000,1.35,35000,1.20,
c 75000,2.0,90000,2.15,130000,1.80,
c table = 'denkt',1,'gdalt',10,1,1,1,
c 0,0.85,10000,0.82,20000,0.80,30000,0.75,35000,0.60,
```

```
c 40000,0.70,100000,0.70,110000,0.60,120000,0.50,130000,0.50,
c table = 'denkt',1,'gdalt',3,1,1,1,
c 0.0,1.0,30000,1.0,130000,1.0,
c $
cl$tab
c table = 'prest',1,'gdalt',6,1,1,1,
c 0,6.52553,1859,6.36383,4217,6.15610,6387,5.96265,8515,5.77084,
c 8816,5.74359,
c $
cl$tab
c table = 'atemt',1,'gdalt',6,1,1,1,
c 0,226.1419,1859,223.766,4217,220.502,6387,217.854,8515,215.292,
c 8816,214.936,
c $
cl$tab
c table = 'vwut',1,'gdalt',6,1,1,1,
c 0,-0.1248,1859,0.02672,4217,0.19915,6387,0.55803,8515,0.86006,
c 8816,0.85601,
c $
cl$tab
c table = 'vwvt',1,'gdalt',6,1,1,1,
c 0,0.200574,1859,0.42458,4217,0.68911,6387,1.5575,8515,2.2659,
c 8816,2.2508,
c $
cl$tab
c table = 'vwwt',0,0.0,
c $
l$tab
 table = 'tvclt',0,2800.0,
 $
l$tab
 table = 'tvc2t',0,2800.0,
 $
l$tab
 table = 'aelt',0,0.2,
 $
l$tab
 table = 'ae2t',0,0.2,
 $
l$tab
 table = 'pilt',0,-90.0,   / engine #1 gimbal pitch angle( reverse x direction)
 $
l$tab
 table = 'yilt',0,0.0,     / engine #1 gimbal yaw angle
 $
l$tab
 table = 'pi2t',0,0.0,    / engine #2 gimbal pitch angle
 $
l$tab
 table = 'yi2t',0,0.0,   / engine #2 gimbal yaw angle
 $
l$tab
 table='cdt',0,1.7,
 $
l$tab
 table='clt',0,0.0,
 $
l$tab
 table='cdplt',0,0.41,
 endphs=1,
 $
c
c Parachute fully deployed
l$gendat
 event=22.,1.,
 critr='diampl',
```

```
  value=13.0,
  parif(1)=0.0,
  wgtsg = 1884.65,/ estimated for MSP'01
  endphs=1,
  $
 l$gendat
  mdl = 1,
 c jetison parachute; turn on engine #1
  event = 50.,
  critr = 'gdalt',
  value = 3500.0,
  wjett = 337.594,/estimated for MSP'01
  sref=2.0,
  iengmf(1) = 1,0,
  iwpf(1) = 1,0,
  $
 l$tblmlt
  $
 l$tab
  table='cdt',0,2.0,
  $
 l$tab
  table='cdplt',0,0.0,
  endphs = 1,
  $
 l$gendat
  mdl = 1,
  event = 70.,
  critr = 'gdalt',
  value = 3000.0,
  endphs = 1,
  $
 c
 c - marks 2500 meter mark (Martian surface)
 l$gendat
  mdl = 1,
  event=80.,
  critr='wr',
  value = 2.0,
  iengmf(1) =  0,0,
  iwpf(1) = 0,0,
  endphs=1,
  $
 c
 c**********************************************************************
 c     This event marks arrival at the  Martian surface
 c**********************************************************************
 l$gendat
  mdl = 1,
  event=500.,
  critr='tdurp',
  value=0,
  endphs=1,
  endjob=1,
  endprb=1,
  $
 l$INPUT_mean
    MONTH     = 02,
    MDAY      = 3,
    MYEAR     = 02,
    NPOS      = 1,
    IHR       = 00,
    IMIN      = 00,
    SEC       = 0.0,
    ALS0      = 0.0,
    INTENS    = 0.0,
```

```
  RADMAX   = 0.0,
  DUSTLAT  = 0.0,
  DUSTLON  = 0.0,
  F107     = 160.0,
  STDL     = 0.0,
  MODPERT  = 3,
  NR1      = 17,
  NVARX    = 2,
  NVARY    = 0,
  LOGSCALE = 0,
  FLAT     = 13.77769,
  FLON     = 269.02228,
  FHGT     = 15.7013,
  DELHGT   = 0.0,
  DELLAT   = 0.0,
  DELLON   = 0.0,
  DELTIME  = 0.0,
  CF0      = 1.00678,
  CF5      = 0.9825277,
  CF15     = 0.9958106,
  CF30     = 0.988124,
  CF50     = 0.857226,
  CF75     = 0.8569739,
  deltaZF  = 0.0,
  deltaTF  = 0.0,
  deltaTEX = 0.0,
  CFp      = 0.725321,
  ipopt    = 1,
$END
```